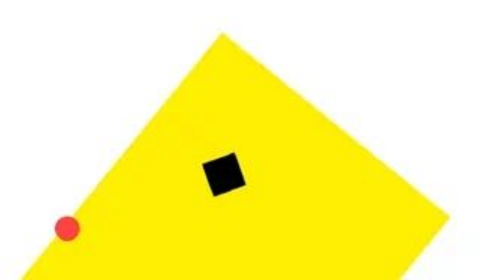
REST поверх WebSocket, используя ServiceWorker API

Антон Петров





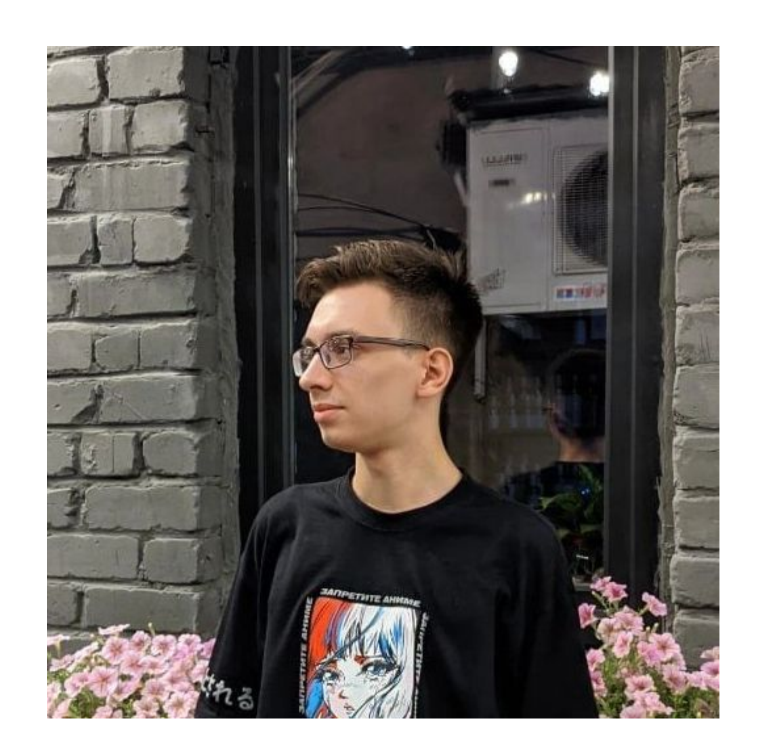
> console.log(self)

Антон Петров
Frontend Developer
Garage Eight

@petrov.engineer

im@petrov.engineer

apetrov@garage-eight.com







> console.log(navigator.vendor)

garage eight

Международная продуктовая IT-компания

Строим экосистему высоконагруженных финансовых продуктов на зарубежных рынках





> throw `too many tabs`





> throw `too many tabs`





> console.log(error.stack)

5 000 × 10 = 50 000



> console.log(error.stack)

~~5 000 × 10 = 50 000~~

5 000 × 10 = 5 000



> solve(this)

beda...

Использование нескольких вкладок с нашим сайтом не поддерживается.
Пожалуйста, закройте все вкладки, кроме одной



> BroadcastChannel

BroadcastChannel

postMessage

message event

message event

window

tab

iframe



> BroadcastChannel

```
const channel = new BroadcastChannel("name");
```



> BroadcastChannel

```
const channel = new BroadcastChannel("name");

channel.addEventListener("message", (event) => {
    console.log(event.data);
});
```



> BroadcastChannel

```
const channel = new BroadcastChannel("name");

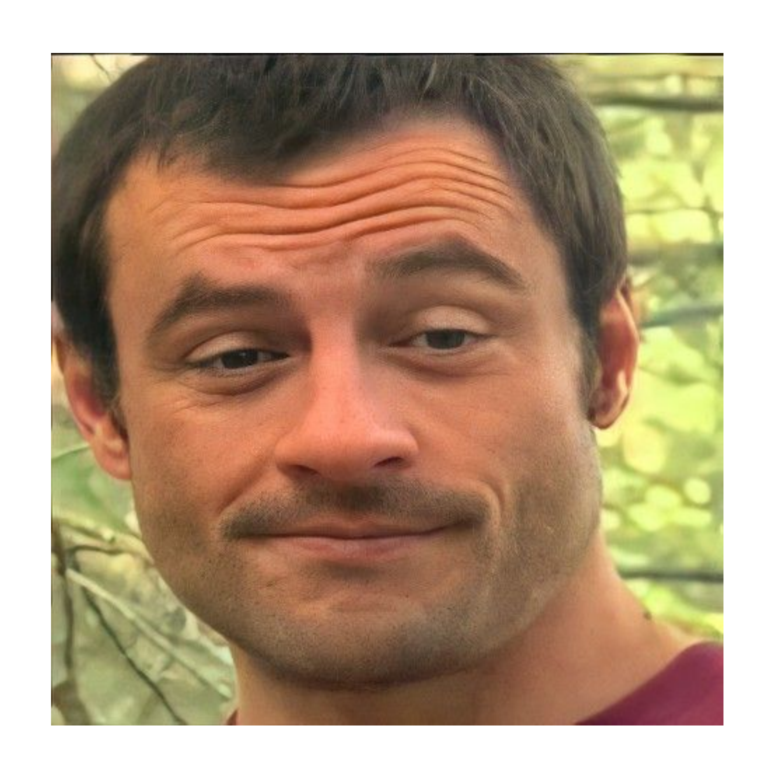
channel.addEventListener("message", (event) => {
    console.log(event.data);
});

channel.postMessage({ frontend: "conf" });
```



> BroadcastChannel

master-slave problem







> SharedWorker

This is not your worker
This is **OUR** worker







> SharedWorker

```
const worker = new SharedWorker("shared.js");
```



> SharedWorker

```
const worker = new SharedWorker("shared.js");

worker.port.start();
```



> SharedWorker

```
const worker = new SharedWorker("shared.js");

worker.port.start();

worker.port.addEventListener("message", (event) => {
   console.log(event.data);
});
```



> SharedWorker

```
const worker = new SharedWorker("shared.js");

worker.port.start();

worker.port.addEventListener("message", (event) => {
   console.log(event.data);
});

worker.port.postMessage({ frontend: "conf" });
```



> SharedWorker

```
addEventListener("connect", (event) => {




});
```



> SharedWorker

```
addEventListener("connect", (event) => {
    const port = event.ports[0];






});
```



> SharedWorker

```
addEventListener("connect", (event) => {
   const port = event.ports[0];

   port.addEventListener("message", (event) => {
       console.log(event.data);
   });


});
```



> SharedWorker

```
addEventListener("connect", (event) => {
  const port = event.ports[0];

  port.addEventListener("message", (event) => {
    console.log(event.data);
  });

  port.start();
});
```



> SharedWorker

Chrome	Edge *	Firefox	Opera
	12-18	2-28	10.1
4-105	79-105	29-104	11.5-90



> SharedWorker

Chrome	Edge *	Firefox	Opera	Safari
				3.1 - 4
	12 - 18	2 - 28	10.1	5 - 6.1
4 - 105	79 - 105	29 - 104	11.5 - 90	7 - 15.6



> console.log(history)

olamedia 2016-07-27 04:25:47 PDT	Comment 4

Can anyone comment what is another option to make shared websocket if you removed this feature?

Lewis 2017-08-22 23:12:46 PDT	Comment 5

SharedWorker is mandatory for implementing the superthread architecture for webapps. Is there any chance that it will be brought back to consideration?

Brady Eidson 2019-10-06 19:52:54 PDT	Comment 6

(In reply to Lewis from comment #5)
> SharedWorker is mandatory for implementing the superthread architecture for
> webapps. Is there any chance that it will be brought back to consideration?

ServiceWorkers can do everything SharedWorkers could. (Or am I wrong about that?)





> ServiceWorker

Cross-Domain Servers / CDNs

Your PWA

Service Worker

Your Server

Client

Proxy Middleware

Servers

User's Device - Browser

Cloud



> ServiceWorker

```
navigator.serviceWorker.register("sw.js");
```



> ServiceWorker

```
navigator.serviceWorker.register("sw.js");

navigator.serviceWorker.addEventListener("message", (event) => {
    console.log(event.data);
});
```



> ServiceWorker

```
navigator.serviceWorker.ready.then((registration) => {
   let worker;
   if (registration.active) worker = registration.active;
   if (registration.waiting) worker = registration.waiting;
   if (registration.installing) worker = registration.installing;
   worker.postMessage("ping");
});
```



> ServiceWorker

```
addEventListener("message", (event) => {
   console.log(event.data);
});
```

> ServiceWorker

```
addEventListener("message", (event) => {
   console.log(event.data);
});

clients.matchAll().then((tabs) => {
   tabs.forEach((tab) => {
       tab.postMessage({ frontend: "conf" });
   });
});
```

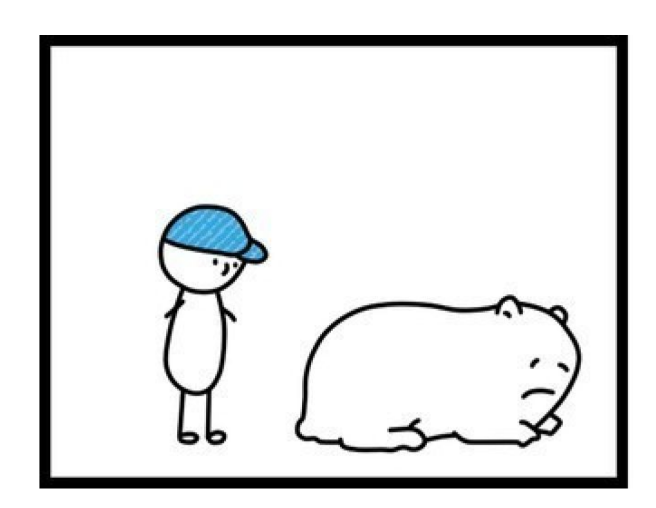
> ServiceWorker

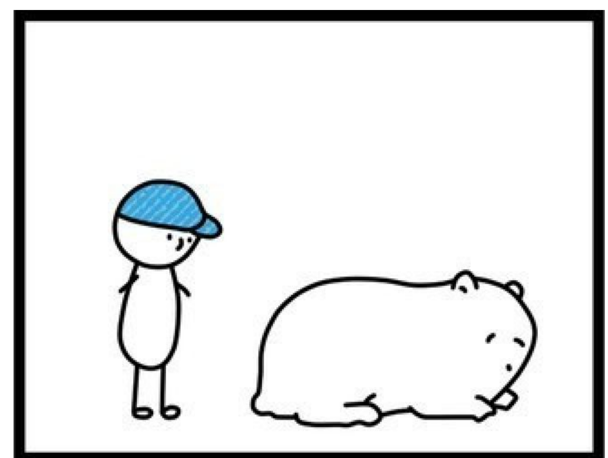
- ✗ BroadcastChannel
- ✗ SharedWorker
- ✓ ServiceWorker



> console.dir(this)

- ping — pong



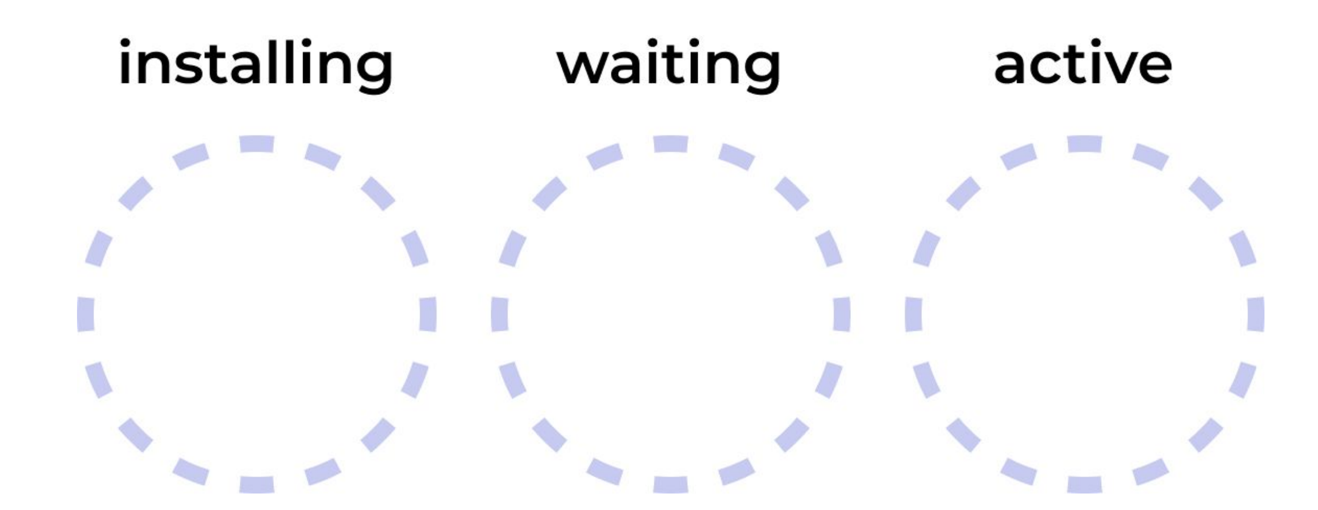






> console.dir(this)

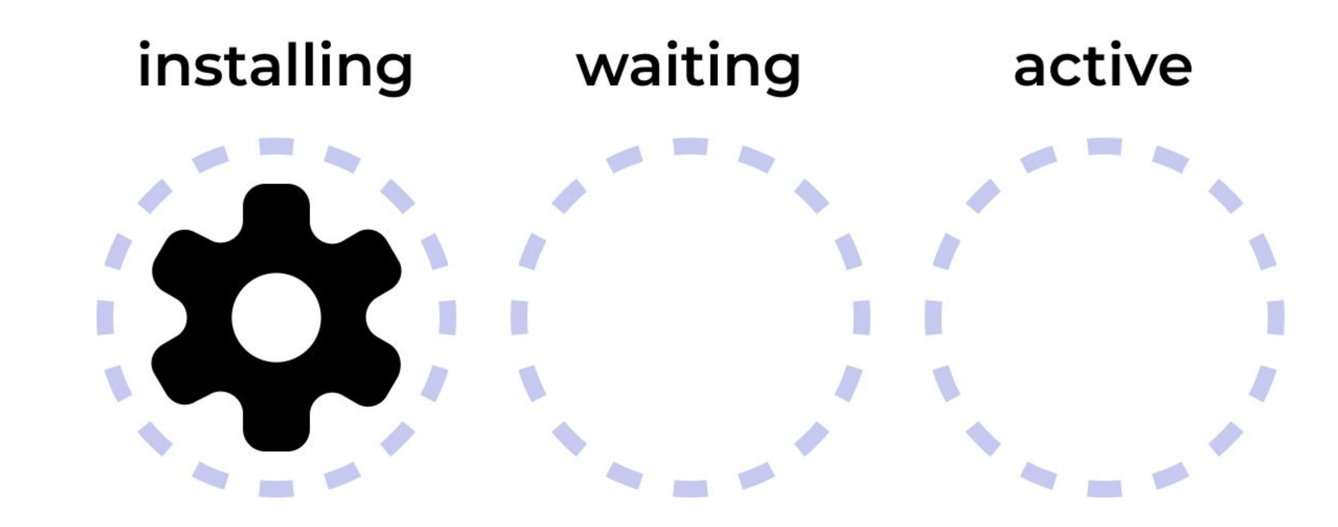
- ping — pong
- обновление





> console.dir(this)

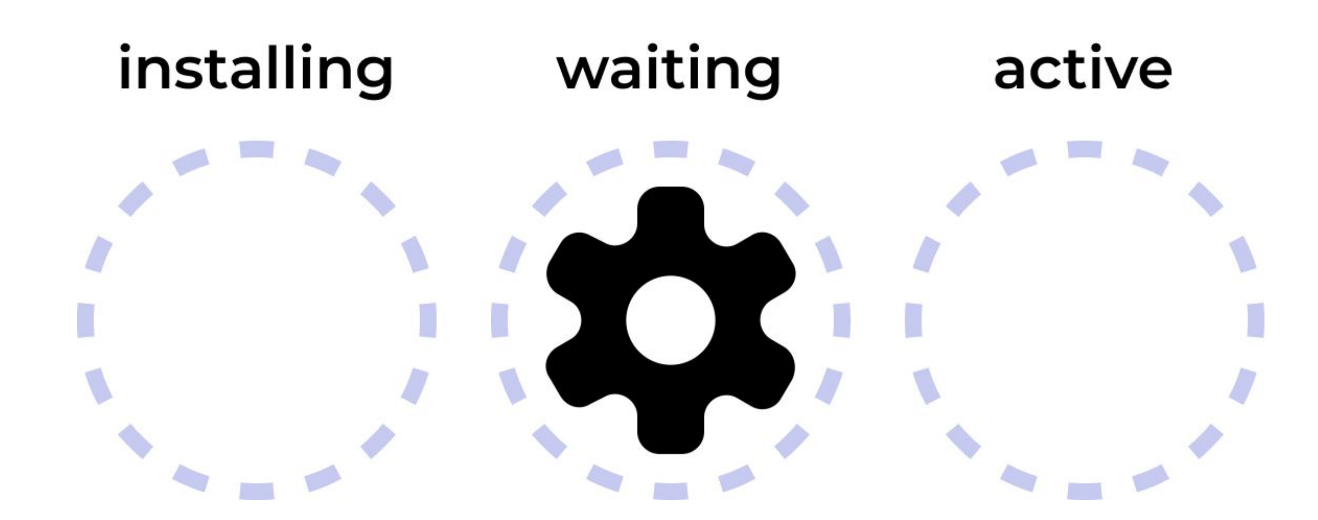
- ping — pong
- обновление





> console.dir(this)

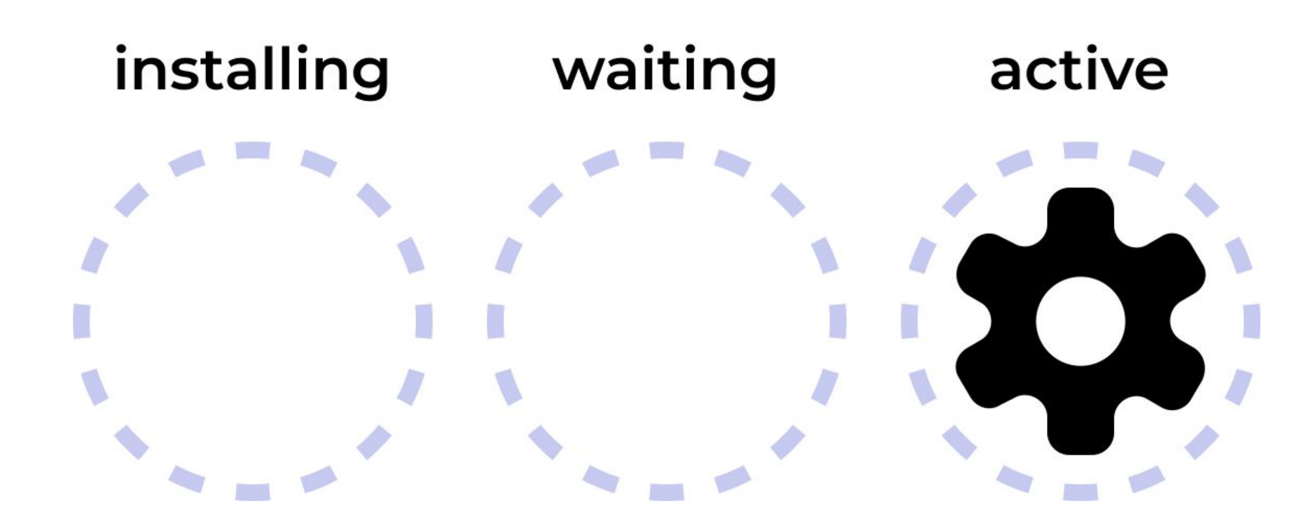
- ping — pong
- обновление





> console.dir(this)

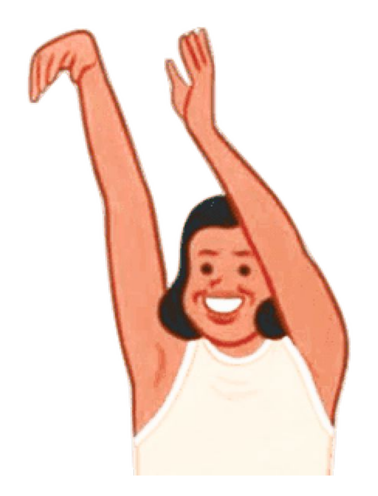
- ping — pong
- обновление





> console.dir(this)





- ping — pong
- обновление
- запрос — ответ





> JSON-RPC 2.0

https://www.jsonrpc.org/specification





> JSON-RPC 2.0

- request

```json
{
    "jsonrpc": "2.0",
    "method": "init",
    "params": {
        "frontend": "conf"
    },
    "id": 1
}
```



> JSON-RPC 2.0

- request
- **notification**

```
{
    "jsonrpc": "2.0",
    "method": "init",
    "params": {
        "frontend": "conf"
    }
}
```



> JSON-RPC 2.0

- request
- notification
- **success**

```json
{
    "jsonrpc": "2.0",
    "result": {
        "year": 2022
    },
    "id": 1
}
```

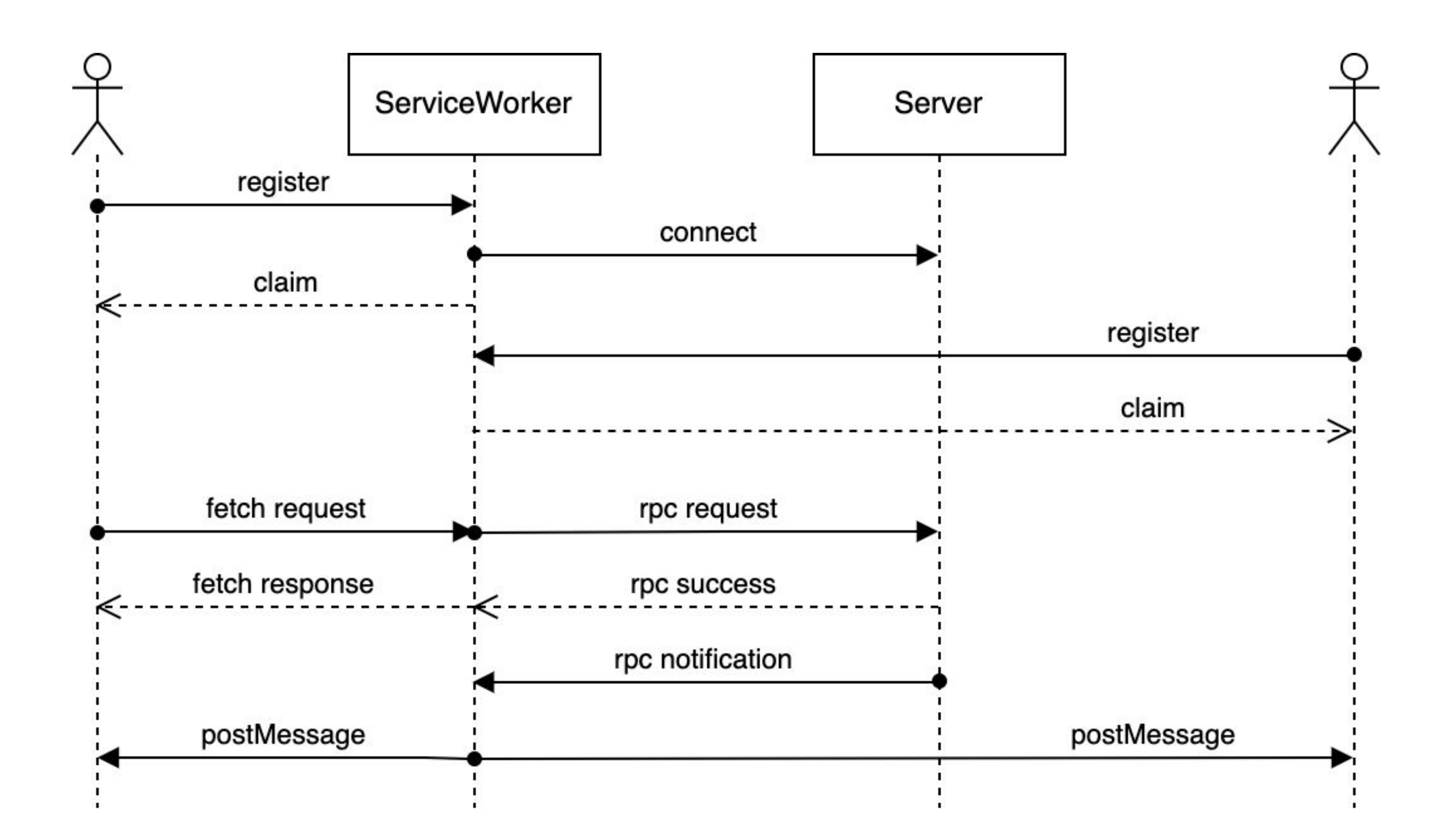


> JSON-RPC 2.0

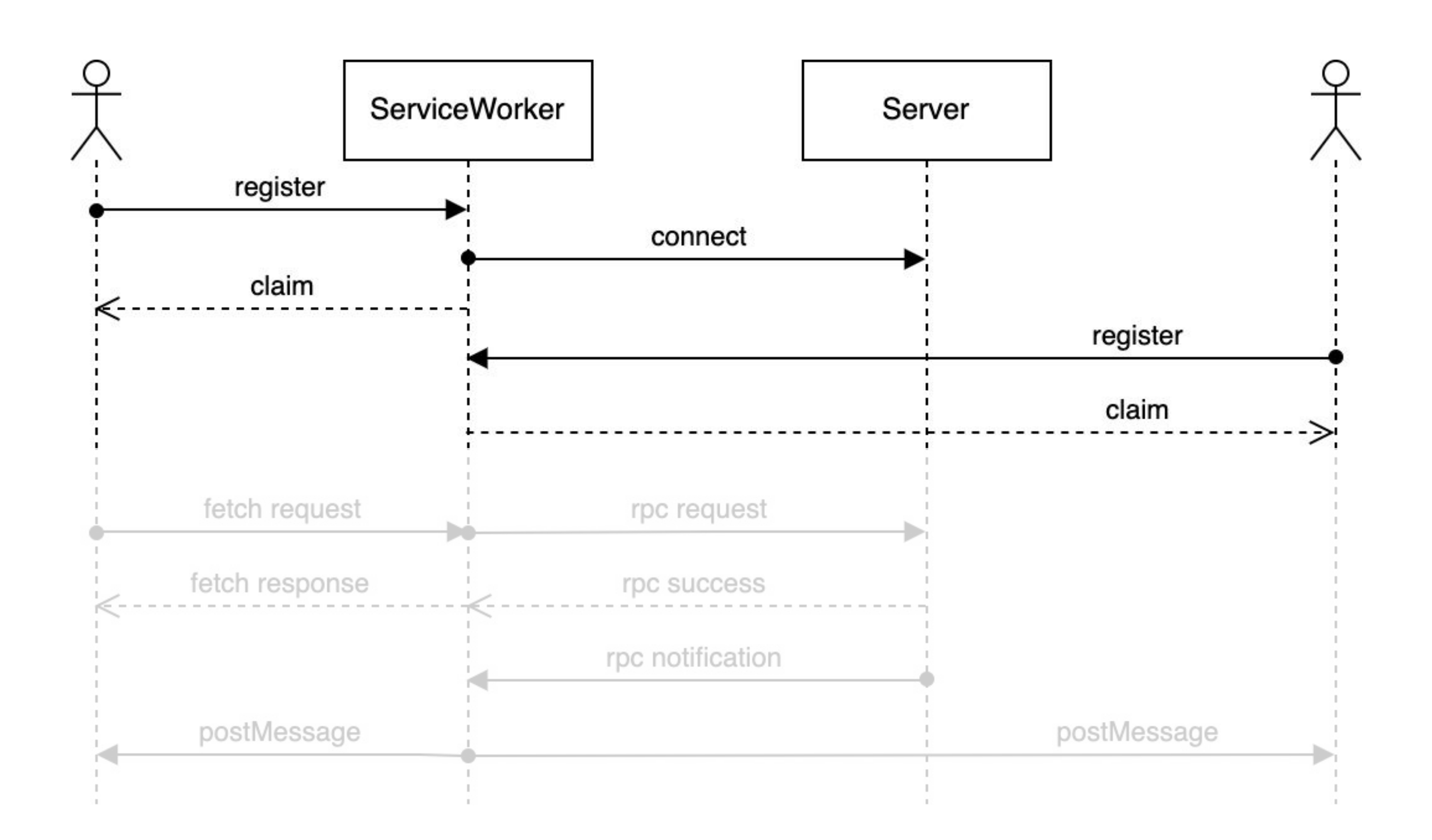
- request
- notification
- success
- **error**

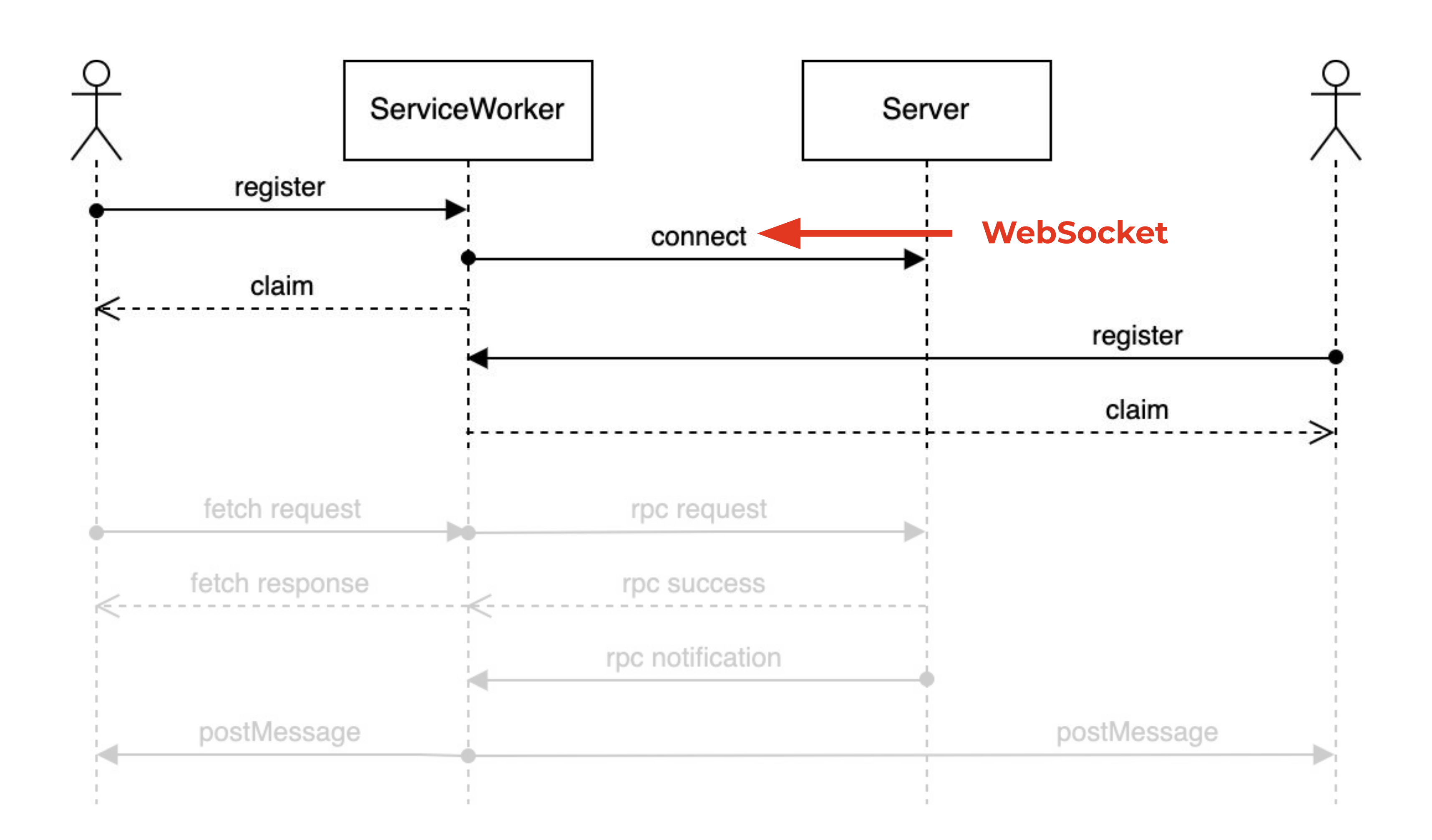
```json
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32000,
        "message": "Server error"
    },
    "id": 1
}
```

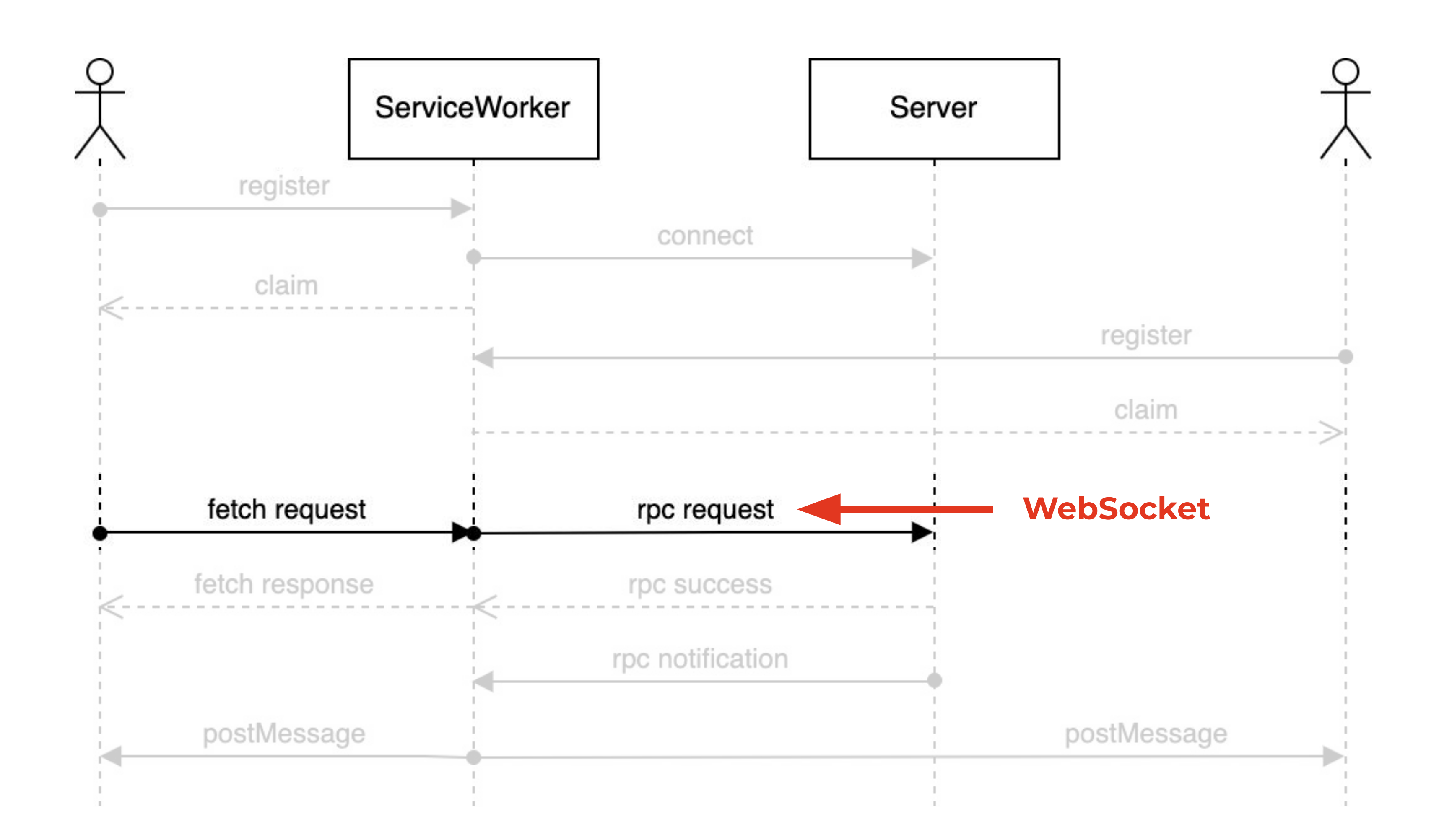


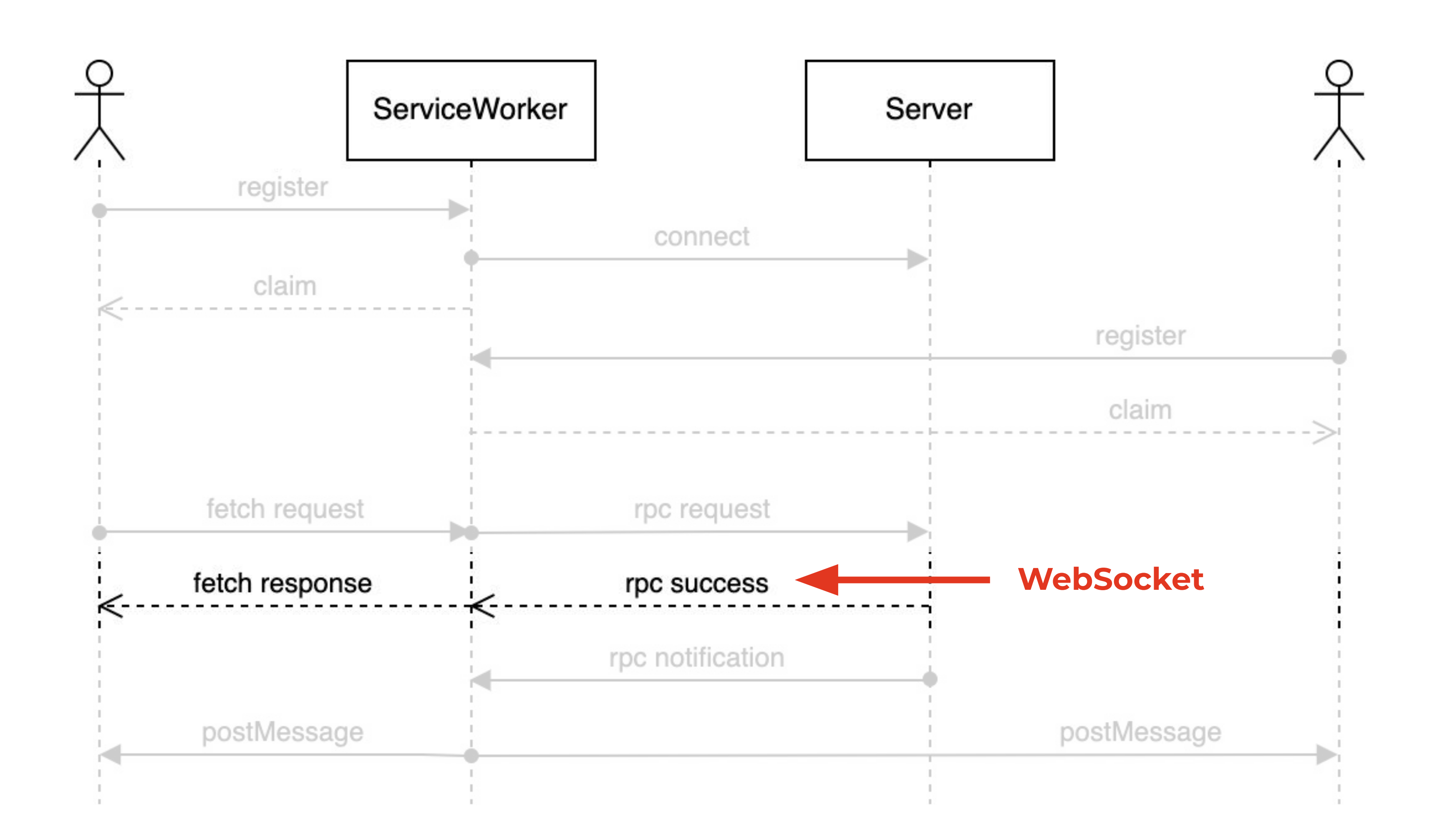


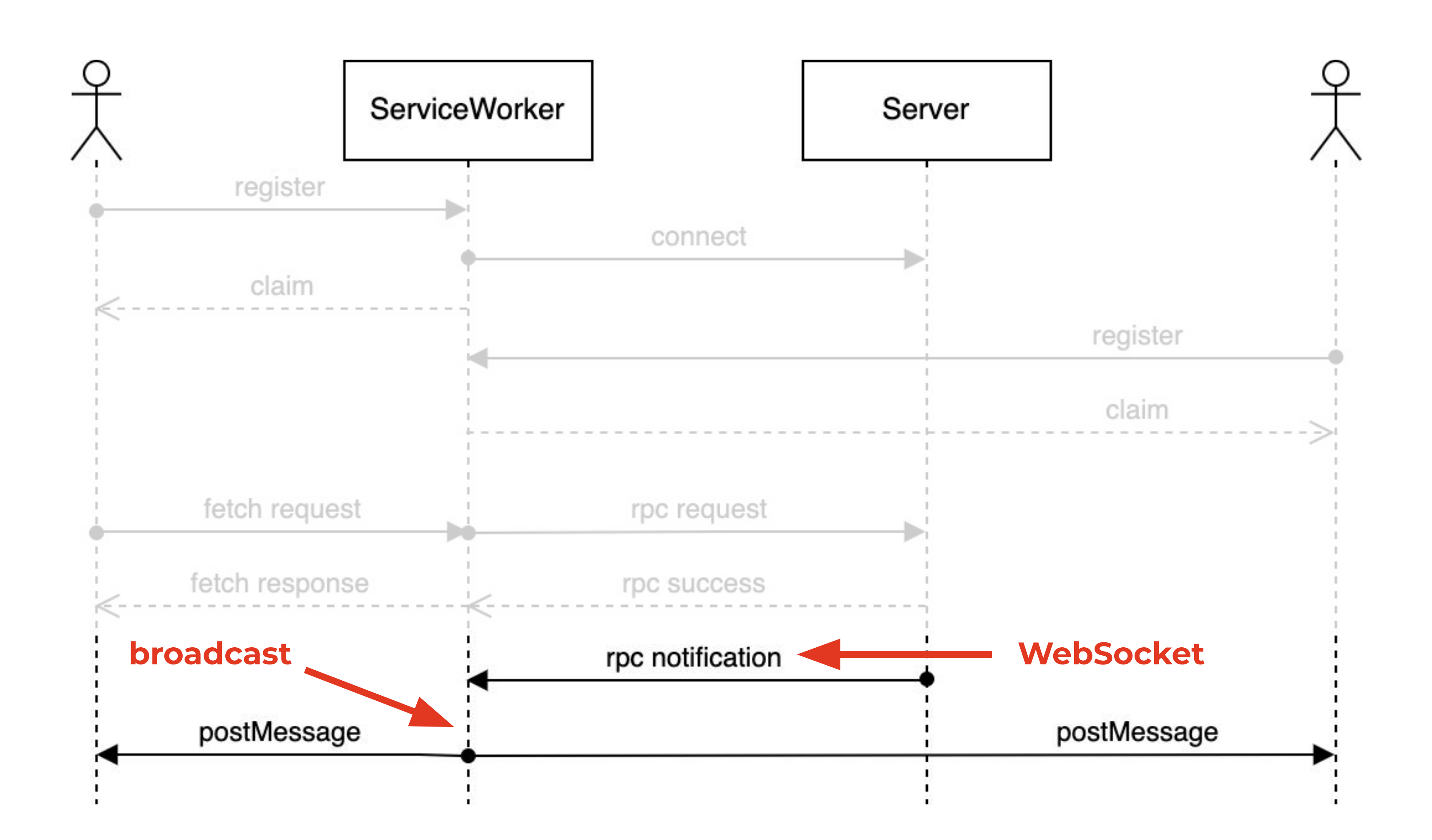
ServiceWorker
Server
register
connect
claim
register
claim
fetch request
rpc request
fetch response
rpc success
rpc notification
postMessage
postMessage


ServiceWorker
Server
register
connect
claim
register
claim
fetch request
rpc request
fetch response
rpc success
rpc notification
postMessage
postMessage


ServiceWorker
Server
register
connect
WebSocket
claim
register
claim
fetch request
rpc request
fetch response
rpc success
rpc notification
postMessage
postMessage

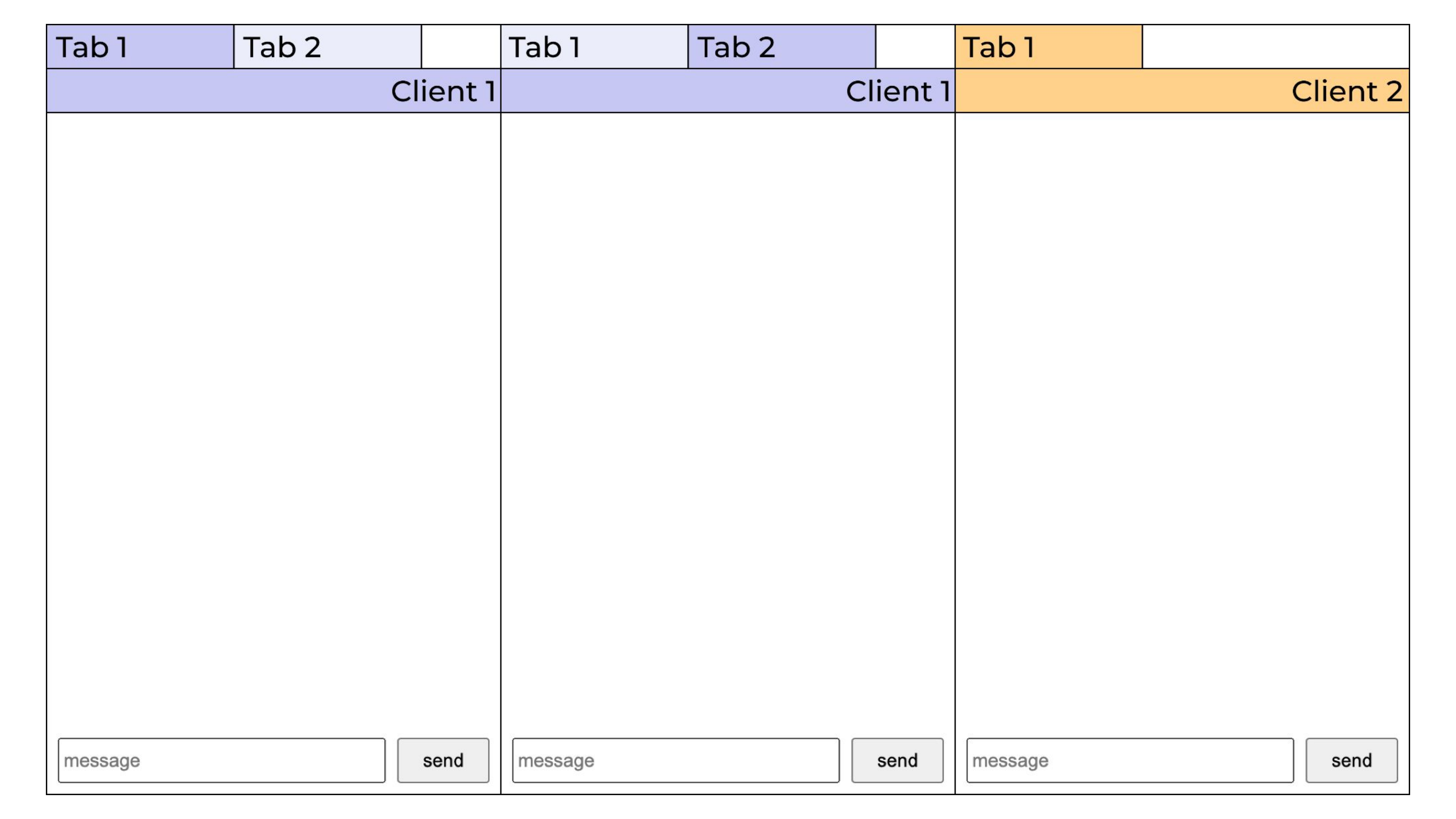

ServiceWorker
Server
register
connect
claim
register
claim
fetch request
rpc request
WebSocket
fetch response
rpc success
rpc notification
postMessage
postMessage

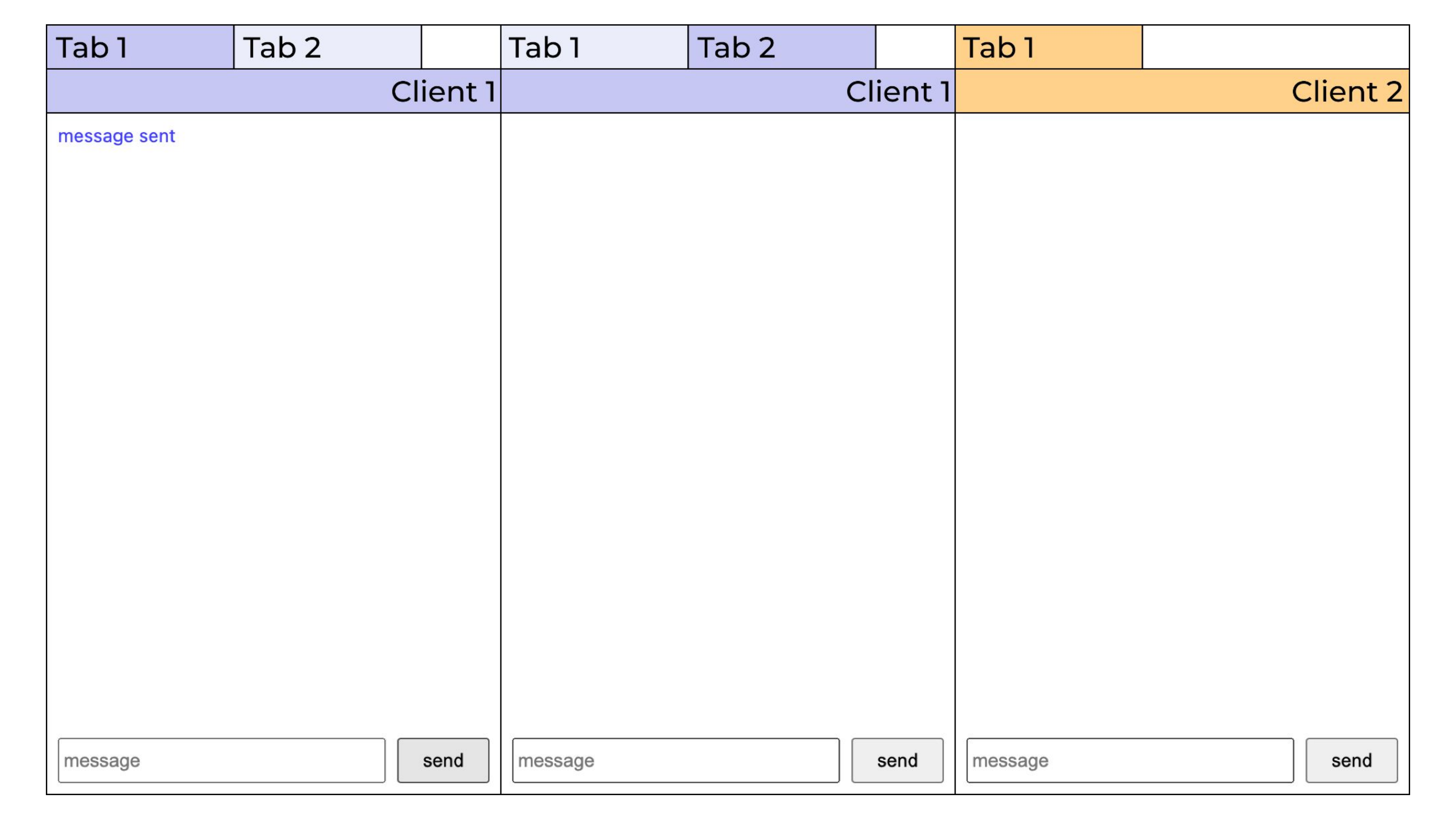

ServiceWorker
Server
register
connect
claim
register
claim
fetch request
rpc request
fetch response
rpc success
WebSocket
rpc notification
postMessage
postMessage


ServiceWorker
Server
register
connect
claim
register
claim
fetch request
rpc request
fetch response
rpc success
broadcast
rpc notification
WebSocket
postMessage
postMessage




FC
Frontend
Conf 2022


Tab 1
Tab 2
Client 1
message
send
Tab 1
Tab 2
Client 1
message
send
Tab 1
Client 2
message
send

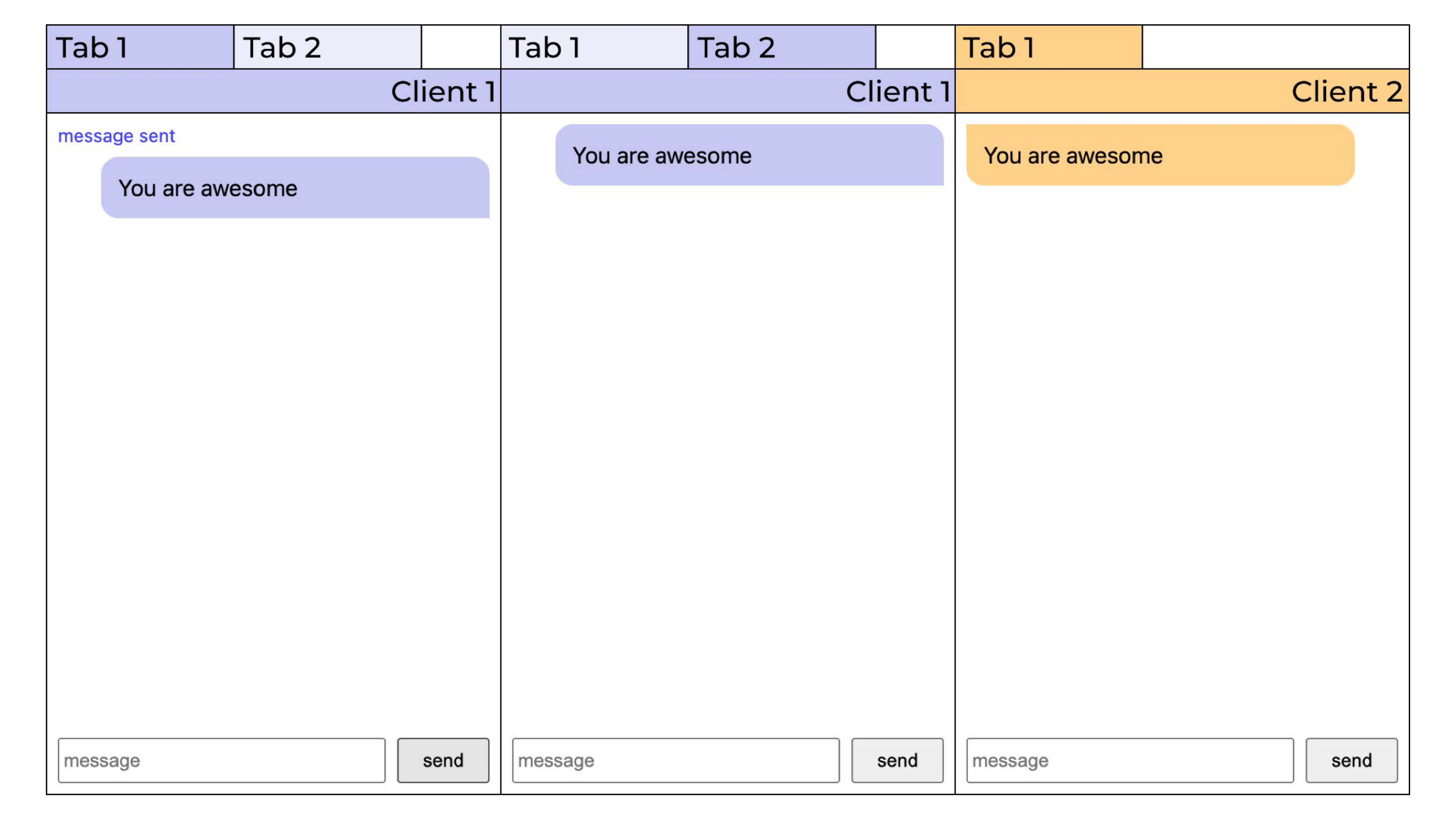


Tab 1
Tab 2
Client 1
message sent
message
send
Tab 1
Tab 2
Client 1
message
send
Tab 1
Client 2
message
send

```javascript
// in tab
fetch("/rpc/message", {
   method: "POST",
   body: JSON.stringify({
       message: "You are awesome"
   })
});

// in worker
addEventListener("fetch", (event) => {
   if (isRPC(event.request.url)) {
       return event.respondWith(rpc(event.request))
   }
});
```

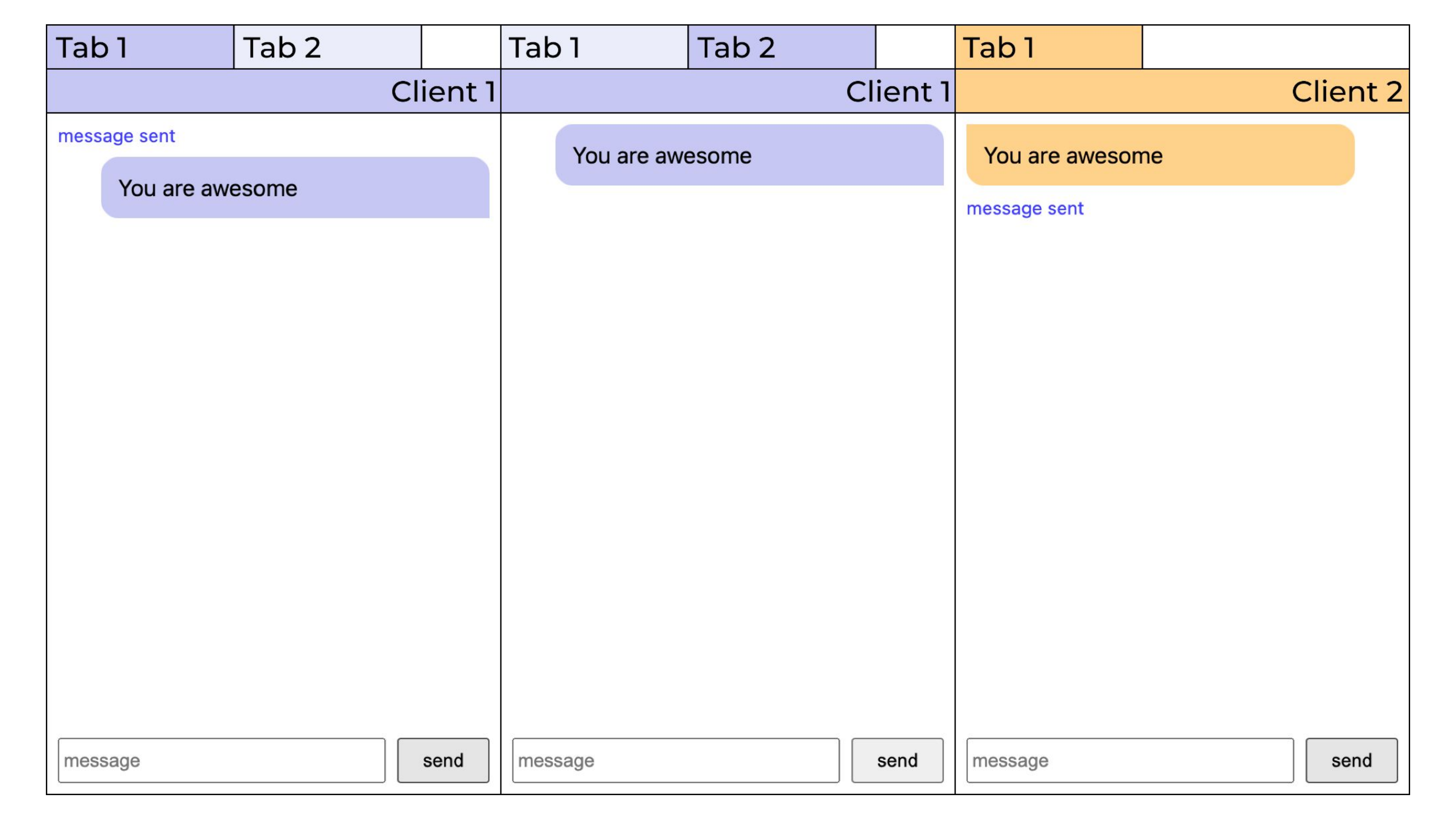


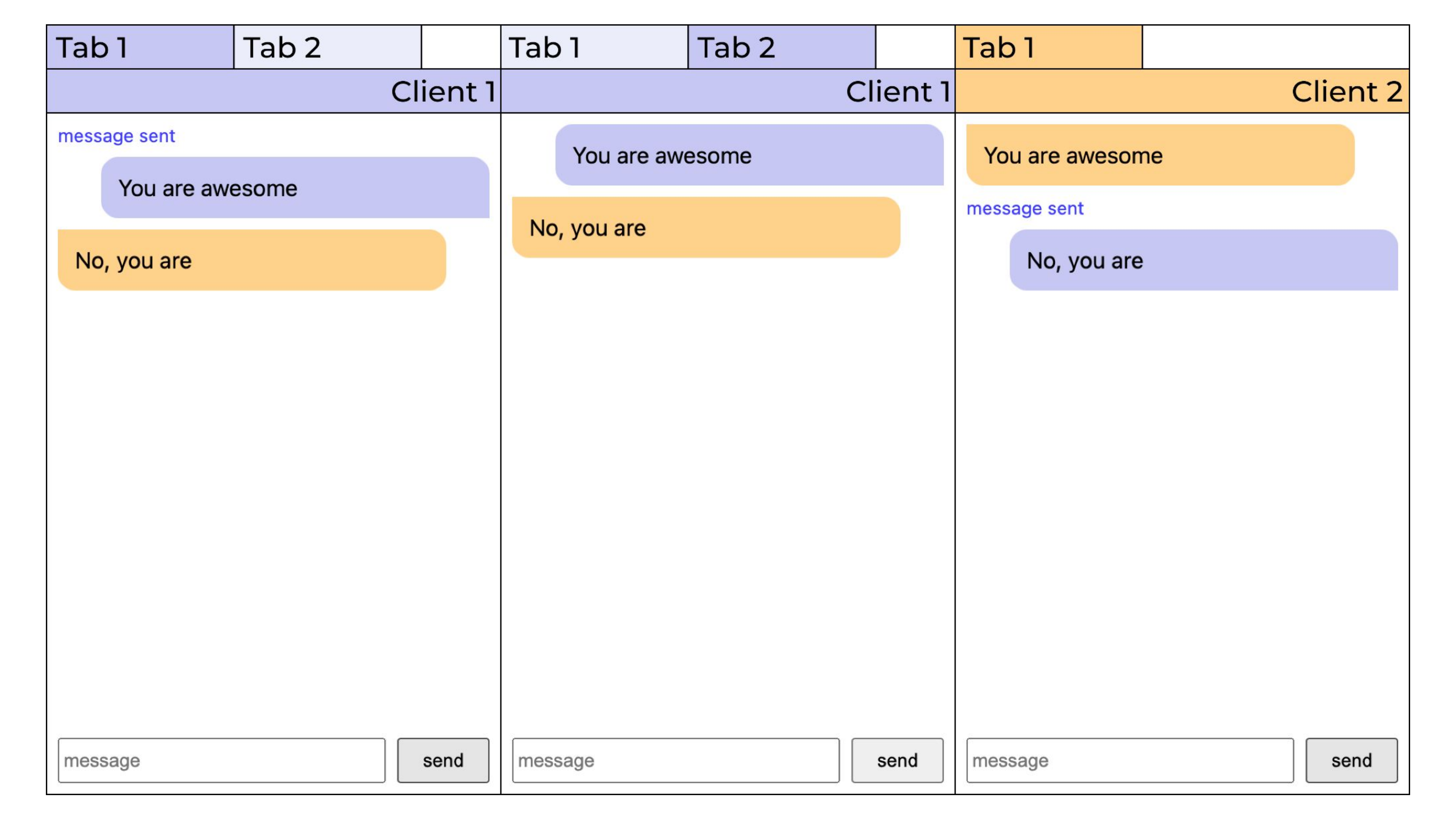

Tab 1
Tab 2
Client 1
message sent
You are awesome
message
send
Tab 1
Tab 2
Client 1
You are awesome
message
send
Tab 1
Client 2
You are awesome
message
send

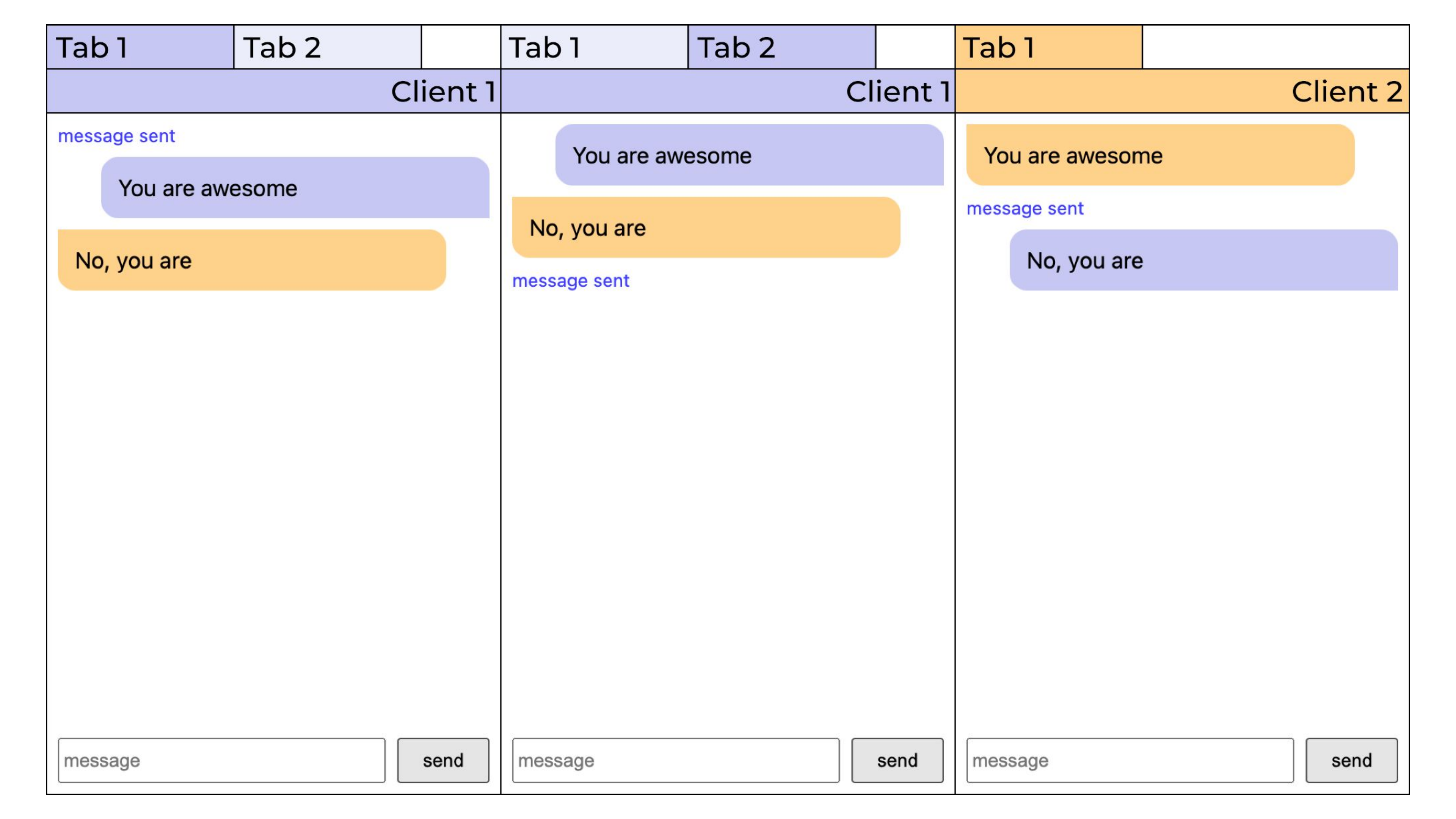
```
const broadcast = (message) => {
  clients.matchAll().then((tabs) => {
    tabs.forEach((tab) => tab.postMessage(message));
  });
};

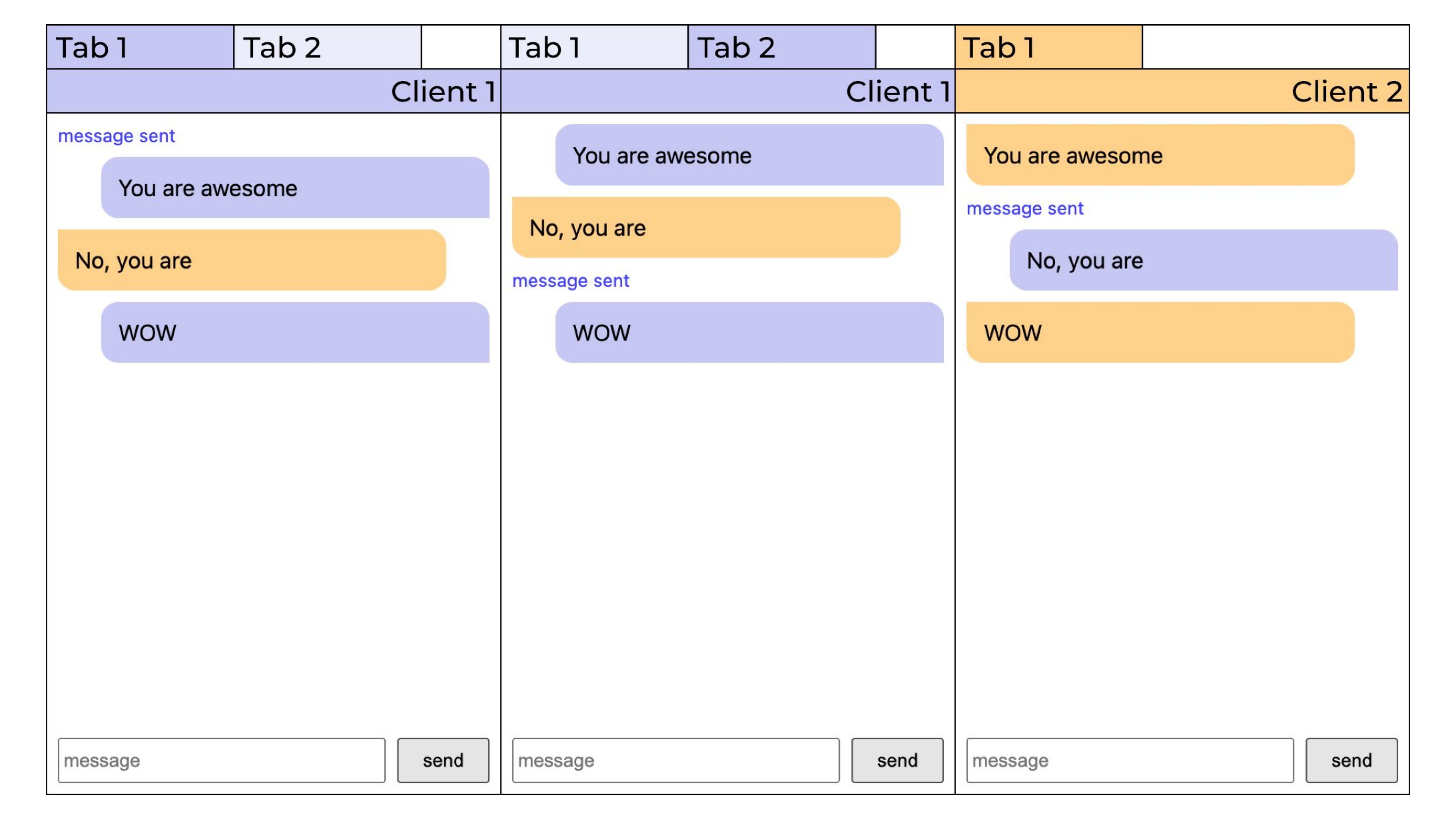
socket.addEventListener("message", (event) => {
  const json = JSON.parse(event.data);
  if (json.method === "message") {
    broadcast(json.params);
  }
});
```




Tab 1
Tab 2
Client 1
message sent
You are awesome
message
send
Tab 1
Tab 2
Client 1
You are awesome
message
send
Tab 1
Client 2
You are awesome
message sent
message
send


Tab 1
Tab 2
Client 1
message sent
You are awesome
No, you are
message
send
Tab 1
Tab 2
Client 1
You are awesome
No, you are
message
send
Tab 1
Client 2
You are awesome
message sent
No, you are
message
send


Tab 1
Tab 2
Client 1
message sent
You are awesome
No, you are
message
send
Tab 1
Tab 2
Client 1
You are awesome
No, you are
message sent
message
send
Tab 1
Client 2
You are awesome
message sent
No, you are
message
send


Tab 1
Tab 2
Client 1
message sent
You are awesome
No, you are
WOW
message
send
Tab 1
Tab 2
Client 1
You are awesome
No, you are
message sent
WOW
message
send
Tab 1
Client 2
You are awesome
message sent
No, you are
WOW
message
send

> navigator.share()

https://github.com/eolme/sw-chat



> exit()

- Многократно снизили нагрузку на Backend



> exit()

- Многократно снизили нагрузку на Backend
- Общение, подобное REST, поверх WebSocket



> exit()

- Многократно снизили нагрузку на Backend
- Общение, подобное REST, поверх WebSocket
- Удобная отладка через DevTools



Кто, если не мы?

@petrov.engineer

im@petrov.engineer

apetrov@garage-eight.com





